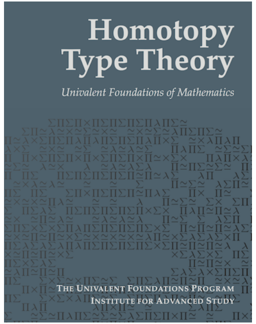
The real numbers in homotopy type theory

Andrej Bauer
University of Ljubljana

Computability and Complexity in Analysis
Faro, Portugal
June 2016

1. Thank you for the invitation.
2. I gave an invited talk at CCA in Hagen, I think it was in 2008, where I spoke about real numbers. Today I am speaking about real numbers again. You know what to expect the next time.

In homotopy type theory there is an *inductive* construction of real numbers.



1. The plan of my talk is to explain a construction of real numbers in homotopy type theory. There is a book, the HoTT book, which explains everything. Chapter 11 is about real numbers.
2. This construction is *inductive*. It gives us *induction* and *recursion* principle for real numbers.
3. Even if you do not care so much for type theory or homotopy type theory, I hope the talk will be interesting because the essential ideas should carry over to other settings, including computable mathematics, although this needs to be properly verified.
4. In addition, the construction is general and applies to other interesting mathematical objects.

Dedekind completeness:

Every cut determines a real.

Cauchy completeness:

Every Cauchy sequence has a limit.

1. There are many ways to construct the real numbers, which can broadly be divided into two classes, according to what kind of completeness we impose on reals.
2. In those mathematical settings where the axiom of countable choice is valid, the difference is inessential as we can show that the two constructions yield the same object (up to a unique isomorphism).
3. Such settings include traditional mathematics, Bishop's constructive mathematics, realizability models (and thus TTE).
4. In (homotopy) type theory, or constructive mathematics without countable choice, the difference matters.
5. In addition, the two notions of completeness lead to different ways of computing with the real numbers. (My talk at CCA 2008 was about computation with Dedekind reals.)
6. The HoTT reals are a Cauchy-style construction.

Construction of Cauchy reals $\mathbb{R}_C$

- Cauchy approximation $a : \mathbb{Q}_+ \to \mathbb{Q}$:

$$\forall \delta, \epsilon \in \mathbb{Q}_+ \,.\, |a_\delta - a_\epsilon| < \delta + \epsilon.$$

- $\mathsf{Cauchy}(\mathbb{Q}) :\equiv \{a : \mathbb{Q}_+ \to \mathbb{Q} \mid a \text{ is C. approx.}\}$.
- Coincidence relation $a \approx b$:

$$a \approx b \iff \forall \delta, \eta \in \mathbb{Q}_+ \,.\, |a_\delta - b_\epsilon| < \delta + \epsilon.$$

- Cauchy reals:

$$\mathbb{R}_C :\equiv \mathsf{Cauchy}(\mathbb{Q})/{\approx}.$$

1. In order to understand what needs to be done, let us take a look at how Cauchy reals are constructed. Let $\mathbb{Q}_+$ be the set of positive rational numbers.
2. We shall use a variant that uses Cauchy approximations instead of sequences. A Cauchy approximation is a map from positive rational numbers which gives for every $\epsilon > 0$ an ϵ-approximation of the limit. Of course, we have to say this without mentioning the limit (which is not there).
3. We complete $\mathbb{Q}$ as a metric space by taking Cauchy approximations, quotiented by the coincidence relation $\approx$. Intuitively "$a \approx b$" means that Cauchy approximations a and b converge to the same limit, but of course we must say this again without mentioning the limit.

Is $\mathbb{R}_C$ Cauchy complete?

- given a Cauchy sequence $(x_k)_k$ of reals,
- every x_i is represented by a Cauchy approximation a_i,
- $\lim_k x_k$ is represented by a Cauchy approximation constructed from the a_i's.

1. Now, how do we know that $\mathbb{R}_C$ is Cauchy complete? It turns out that the natural way of proving completeness uses countable choice! The root cause of this is the fact that taking the limit of a Cauchy sequence is an *infinitary* operation.
2. Can we avoid the use of countable choice?

Is $\mathbb{R}_C$ Cauchy complete?

- given a Cauchy sequence $(x_k)_k$ of reals,
- every x_i is represented by a Cauchy approximation a_i,
- $\lim_k x_k$ is represented by a Cauchy approximation constructed from the a_i's.

We used countable choice to get from x_i to a_i!

1. Now, how do we know that $\mathbb{R}_C$ is Cauchy complete? It turns out that the natural way of proving completeness uses countable choice! The root cause of this is the fact that taking the limit of a Cauchy sequence is an *infinitary* operation.
2. Can we avoid the use of countable choice?

Completion of completion of …

- Let $\overline{M} :\equiv \mathsf{Cauchy}(M)/{\approx}$.
- Iterate the process of metric completion?

$$\mathbb{Q} \subseteq \overline{\mathbb{Q}} \subseteq \overline{\overline{\mathbb{Q}}} \subseteq \overline{\overline{\overline{\mathbb{Q}}}} \subseteq \cdots$$

1. Let us write $\overline{M}$ for the metric completion of a metric space M, i.e., $\overline{M}$ is the space of Cauchy approximations in M, quotiented by a coincidence relation $\approx$.
2. Since $\mathbb{R}_C$, which is just $\overline{\mathbb{Q}}$, does not seem to be complete, we could do one more step, and another one, and so on. We would *not* be done after ω steps and we would have to iterate into transfinite ordinals. This is quite nasty, as constructively the ordinals are not behaved nicely.
3. A second attempt might use the fact that the Dedekind reals $\mathbb{R}_D$ *are* Cauchy complete, because every Cauchy sequence determines a Dedekind cut. So, we could get $\mathbb{R}_C$ as the least Cauchy-complete subfield of $\mathbb{R}_D$. However, this presupposes that we have $\mathbb{R}_D$!
4. Fred Richman has a nice construction of metric completions without countable choice. He uses powersets, and so his construction is not available in type theory either.

Completion of completion of . . .

- Let $\overline{M} :\equiv \mathsf{Cauchy}(M)/\approx$.
- Iterate the process of metric completion?

$$\mathbb{Q} \subseteq \overline{\mathbb{Q}} \subseteq \overline{\overline{\mathbb{Q}}} \subseteq \overline{\overline{\overline{\mathbb{Q}}}} \subseteq \cdots \subseteq \overline{\mathbb{Q}}^{(\omega)} \subseteq \overline{\overline{\mathbb{Q}}^{(\omega+1)}} \subseteq \cdots$$

1. Let us write $\overline{M}$ for the metric completion of a metric space M, i.e., $\overline{M}$ is the space of Cauchy approximations in M, quotiented by a coincidence relation $\approx$.
2. Since $\mathbb{R}_C$, which is just $\overline{\mathbb{Q}}$, does not seem to be complete, we could do one more step, and another one, and so on. We would *not* be done after ω steps and we would have to iterate into transfinite ordinals. This is quite nasty, as constructively the ordinals are not behaved nicely.
3. A second attempt might use the fact that the Dedekind reals $\mathbb{R}_D$ *are* Cauchy complete, because every Cauchy sequence determines a Dedekind cut. So, we could get $\mathbb{R}_C$ as the least Cauchy-complete subfield of $\mathbb{R}_D$. However, this presupposes that we have $\mathbb{R}_D$!
4. Fred Richman has a nice construction of metric completions without countable choice. He uses powersets, and so his construction is not available in type theory either.

Completion of completion of ...

- Let $\overline{M} :\equiv \mathsf{Cauchy}(M)/\approx$.
- Iterate the process of metric completion?

$$\mathbb{Q} \subseteq \overline{\mathbb{Q}} \subseteq \overline{\overline{\mathbb{Q}}} \subseteq \overline{\overline{\overline{\mathbb{Q}}}} \subseteq \cdots \subseteq \overline{\mathbb{Q}}^{(\omega)} \subseteq \overline{\mathbb{Q}}^{(\omega+1)} \subseteq \cdots$$

- The Cauchy completion of $\mathbb{Q}$ is the least Cauchy complete subfield of the Dedekind reals $\mathbb{R}_D$.
  - ...but this presupposes we have $\mathbb{R}_D$.

1. Let us write $\overline{M}$ for the metric completion of a metric space M, i.e., $\overline{M}$ is the space of Cauchy approximations in M, quotiented by a coincidence relation $\approx$.
2. Since $\mathbb{R}_C$, which is just $\overline{\mathbb{Q}}$, does not seem to be complete, we could do one more step, and another one, and so on. We would *not* be done after ω steps and we would have to iterate into transfinite ordinals. This is quite nasty, as constructively the ordinals are not behaved nicely.
3. A second attempt might use the fact that the Dedekind reals $\mathbb{R}_D$ *are* Cauchy complete, because every Cauchy sequence determines a Dedekind cut. So, we could get $\mathbb{R}_C$ as the least Cauchy-complete subfield of $\mathbb{R}_D$. However, this presupposes that we have $\mathbb{R}_D$!
4. Fred Richman has a nice construction of metric completions without countable choice. He uses powersets, and so his construction is not available in type theory either.

The inductive type $\mathsf{Tree}(A)$

Constructors:

- $\mathsf{leaf} : A \to \mathsf{Tree}(A)$
- $\mathsf{tree} : \mathsf{Tree}(A) \times \mathsf{Tree}(A) \to \mathsf{Tree}(A)$

1. To explain how we are going to overcome the difficulties, we have to make an excursion into the topic of inductive definitions. A prototypical inductive definition is that of (non-empty) finite binary trees over a set A.
2. First we have constructors, which can be used to construct trees.
3. Then there is the induction principle. It unifies both the usual induction principle for proving properties of trees, and a recursion principle for defining functions by recursion on trees.
4. As a logical principle it reads as shown here.

The inductive type $\mathsf{Tree}(A)$

Constructors:

- $\mathsf{leaf} : A \to \mathsf{Tree}(A)$
- $\mathsf{tree} : \mathsf{Tree}(A) \times \mathsf{Tree}(A) \to \mathsf{Tree}(A)$

Induction principle:

$$
\begin{aligned}
\mathsf{ind}_{\mathsf{Tree}(A)} : \ & \textstyle\prod_{P:A\to\mathsf{Type}} (\prod_{a:A} P(\mathsf{leaf}(a))) \to \\
& (\textstyle\prod_{x,y:\mathsf{Tree}(A)} P(x) \times P(y) \to P(\mathsf{tree}(x,y))) \to \\
& \textstyle\prod_{z:\mathsf{Tree}(A)} P(z)
\end{aligned}
$$

1. To explain how we are going to overcome the difficulties, we have to make an excursion into the topic of inductive definitions. A prototypical inductive definition is that of (non-empty) finite binary trees over a set A.
2. First we have constructors, which can be used to construct trees.
3. Then there is the induction principle. It unifies both the usual induction principle for proving properties of trees, and a recursion principle for defining functions by recursion on trees.
4. As a logical principle it reads as shown here.

The inductive type $\mathsf{Tree}(A)$

Constructors:

- $\mathsf{leaf} : A \to \mathsf{Tree}(A)$
- $\mathsf{tree} : \mathsf{Tree}(A) \times \mathsf{Tree}(A) \to \mathsf{Tree}(A)$

Induction principle:

- if P is a property of trees such that:
  - for all $a \in A$, $P(\mathsf{leaf}(a))$
  - for all $x, y \in \mathsf{Tree}(A)$,
    if $P(x)$ and $P(y)$ then $P(\mathsf{tree}(x, y))$
- then $P(z)$ for all $z \in \mathsf{Tree}(A)$.

1. To explain how we are going to overcome the difficulties, we have to make an excursion into the topic of inductive definitions. A prototypical inductive definition is that of (non-empty) finite binary trees over a set A.
2. First we have constructors, which can be used to construct trees.
3. Then there is the induction principle. It unifies both the usual induction principle for proving properties of trees, and a recursion principle for defining functions by recursion on trees.
4. As a logical principle it reads as shown here.

Recursion principle for $\mathsf{Tree}(A)$

Given

- a type B
- a map $\ell : A \to B$
- a map $\tau : \mathsf{Tree}(A) \times \mathsf{Tree}(A) \times B \times B \to B$

there is $r : A \to B$ such that

$$\begin{aligned} r(\mathsf{leaf}(a)) &= \ell(a) \\ r(\mathsf{tree}(x, y)) &= \tau(x, y, r(x), r(y)) \end{aligned}$$

1. The recursion principle is obtained by taking $P(x) = B$ for a fixed type B, as shown.
2. You should recognize here the usual folding operation on trees. The equations are called *computation rules*. Of course, there are analogous computation rules for the full induction principle.
3. Henceforth we shall only look at recursion principles, because they are simpler. But keep in mind that in type theory inductive definitions have a general inductive principle, of which the recursion principle is a special case.

Finite sets $\mathsf{Fin}(A)$

Constructors:

- $\{-\} : A \to \mathsf{Fin}(A)$
- $\cup : \mathsf{Fin}(A) \times \mathsf{Fin}(A) \to \mathsf{Fin}(A)$

Equations:

- $x \cup x = x$,
- $x \cup y = y \cup x$,
- $(x \cup y) \cup z = x \cup (y \cup z)$.

for all $x, y, z \in \mathsf{Fin}(A)$.

1. The inductive types are very useful as a programming device, but in mathematics we need inductive definitions *with equations*. To draw a parallel with finite trees, let us consider the (non-empty) finite subsets of a given set A, expressed as an inductive type with equations.
2. First we have two constructors, for forming singleton subsets and binary unions. They correspond respectively to the leaves and the composite trees from the previous example.
3. The unions satisfy equations: idempotency, commutativity and associativity.
4. The usual way to deal with the equations is to *quotient* by the equivalence relation generated by them. But this leads to problems with axiom of choice as soon as the constructors are infinitary, like the limit of a Cauchy sequence. We need to do something else.

Finite sets $\mathsf{Fin}(A)$ as a HIT

Constructors:

- $\{-\} : A \to \mathsf{Fin}(A)$
- $\cup : \mathsf{Fin}(A) \times \mathsf{Fin}(A) \to \mathsf{Fin}(A)$

Path constructors:

- $\mathsf{idem} : \prod_{x:\mathsf{Fin}(A)} \mathsf{Id}(x \cup x, x)$,
- $\mathsf{comm} : \prod_{x,y:\mathsf{Fin}(A)} \mathsf{Id}(x \cup y, y \cup x)$,
- $\mathsf{assoc} : \prod_{x,y,z:\mathsf{Fin}(A)} \mathsf{Id}((x \cup y) \cup z, x \cup (y \cup z))$.

for all $x, y, z \in \mathsf{Fin}(A)$.

1. We shall use a *higher inductive type*. This is an inductive type in which the desired equalities are expressed as *path constructors*.
2. We can explain this at an intuitive level as follows. A set has elements, while a type has not only elements but also *paths* connecting them. This is the homotopy-theoretic understanding of types. And so, if we want to make two elements equal, we just put in a path.
3. All that is needed to complete the definition is a suitable induction principle. Let us look at the derived recursion principle that corresponds to initiality of $\mathsf{Fin}(A)$.

Recursion principle for $\mathsf{Fin}(A)$

Given

- a type B,
- $s : A \to B$,
- $u : B \times B \to B$,
- $\alpha : \prod_{x:B} \mathsf{Id}(u(x,x), x)$
- $\beta : \prod_{x,y:B} \mathsf{Id}(u(x,y), u(y,x))$
- $\gamma : \prod_{x,y,z:B} \mathsf{Id}(u(x, u(y,z)), u(u(x,y),z))$

Then there is a map $r : \mathsf{Fin}(A) \to B$ such that

$$r(\{a\}) = s(a), \qquad r(x \cup y) = u(r(x), r(y)).$$

1. The premises say that (B, u) is a semilattice “up to path equality”, as witnessed by α, β, and γ. The map s explains how to map the generators into B.
2. We are still lacking uniqueness of r, as well as the fact that it is a homomorphism. These are obtained from suitable computation rules for r, which we omit here. See the HoTT book, Section 6.11, for details.

Truncation of paths

Truncation:

- trunc : $\prod_{x,y:\mathsf{Fin}(A)} \prod_{p,q:\mathsf{Id}(x,y)} \mathsf{Id}(p,q)$

1. There is at this point a further technical issue which we should mention.
2. When we introduce new paths between elements, they can be composed and inverted, so we actually get many more paths. It can happen that we have *too many* new paths that create homotopically non-trivial phenomena.
3. In order to get the usual behavior of initial algebras, we have to trivialize the higher homotopical structure. This is done by putting in even more paths: for any two parallel paths p and q there is a (two-dimensional) path between them. Luckily, we do not have to put in even more higher-dimensional paths after that.
4. Henceforth all our constructions will implicitly include the truncation and we shall not mention it again.

$\mathbb{R}_C$ as a higher inductive-inductive type

- Constructors:
  - rational points
  - limit points
- Path constructor
- Truncation
- Induction principle
- Auxiliary relation $x \sim_\epsilon y$ ("x and y are ϵ-close").

1. We are ready to define the Cauchy reals as a higher inductive type. Let us summarize how this is going to work.
2. There are two kinds of constructors for creating reals, one for rational points and another for limit points.
3. There is a path constructor for any two reals which are ϵ-close for all $\epsilon : \mathbb{Q}_+$.
4. There is also the truncation of paths, and the induction principle, but we are not going to look at those.
5. In order to explain what it means for x and y to be ϵ-close we need an auxiliary proximity relation $x \sim_\epsilon y$. We cannot define this as $|x - y| < \epsilon$ because we do not yet have any arithmetic and order. Instead, we define $\sim_\epsilon$ inductively and simultaneously with $\mathbb{R}_C$.
6. This is known as a higher-inductive-inductive type. The HoTT book contains other examples, for instance models of Zermelo-Fraenkel set theory.

Point and path constructors in $\mathbb{R}_C$

- Point constructors:
  - *rational points*: for every $q \in \mathbb{Q}$ there is $\mathsf{rat}(q) : \mathbb{R}_C$
  - *limit points*: for every $x : \mathbb{Q}_+ \to \mathbb{R}_C$ such that

    $$\prod_{\delta,\epsilon:\mathbb{Q}_+} x_\delta \sim_{\delta+\epsilon} x_\epsilon$$

    there is $\mathsf{lim}(x) : \mathbb{R}_C$ (x is a *Cauchy approximation*).
- Path constructor: for all $u, v : \mathbb{R}_C$ such that

  $$\prod_{\epsilon:\mathbb{Q}_+} u \sim_\epsilon v$$

  there is a path $\mathsf{eq}(u, v) : \mathsf{Id}(u, v)$.

1. There are two path constructors, one for rational points and another for Cauchy approximation. Please note that lim is a formal symbol here, it does not actually compute anything. (You don't think that $\mathsf{rat}(q)$ is about rodents, do you?)
2. There is a path constructor which inserts a path from u to v whenever u and v are ϵ-close for all $\epsilon : \mathbb{Q}_+$.

Auxiliary relation $\sim_\epsilon$

Define inductively:

- for any q, r, ϵ, if $|q - r| < \epsilon$ then $\mathsf{rat}(q) \sim_\epsilon \mathsf{rat}(r)$
- for any q, y, ϵ, δ, if $\mathsf{rat}(q) \sim_{\epsilon-\delta} y_\delta$ then $\mathsf{rat}(q) \sim_\epsilon \mathsf{lim}(y)$
- for any x, r, ϵ, δ, if $x_\delta \sim_{\epsilon-\delta} \mathsf{rat}(r)$ then $\mathsf{lim}(x) \sim_\epsilon \mathsf{rat}(r)$
- for any $x, y, \epsilon, \delta, \eta$, if $x_\delta \sim_{\epsilon-\delta-\eta} y_\eta$ then $\mathsf{lim}(x) \sim_\epsilon \mathsf{lim}(y)$
- for any u, v, ϵ, if $\zeta, \xi : u \sim_\epsilon v$ then $\mathsf{Id}(\zeta, \xi)$ (propositional truncation)

1. The auxiliary relation is defined inductively by four clauses, one for each combination of point constructors.
2. The first clause is the base case: for rational numbers $\sim_\epsilon$ actually corresponds to ϵ-proximity.
3. The second and third clauses explain when a rational is ϵ-close to a limit point.
4. The fourth clause relates two limit points.
5. The last clause is a technicality. It inserts a path between any two elements of $u \sim_\epsilon v$ to make sure that $\sim_\epsilon$ is a *proposition* (has at most one element).

$\mathbb{R}_C$-recursion

To construct a map $f : \mathbb{R}_C \to B$ by recursion:

- for every $q : \mathbb{Q}$ construct an element $f(\mathsf{rat}(q)) : B$,
- for every Cauchy approximation $x : \mathbb{Q}_+ \to \mathbb{R}_C$, construct $f(x) : B$, assuming $f(x_\epsilon) : B$ has been defined already for all $\epsilon : \mathbb{Q}_+$,
- for all $u, v : \mathbb{R}_C$ if $\prod_{\epsilon:\mathbb{Q}_+} u \sim_\epsilon v$, a path $\mathsf{Id}(f(u), f(b))$.

1. There are several kind of induction and recursion principles that can be derived from the general induction principle for $\mathbb{R}_C$. Here is a very simple one, which is not the most useful one. Please consult the HoTT book for more information.
2. The recursion principle follows a general pattern. For every point constructor we give the value of the recursive function f, assuming f has already been defined on the recursive arguments.
3. In addition, for every path constructor we need to show that the function f respects the paths.

Theorem: $\mathbb{R}_C$ *is the initial Cauchy-complete field.*

1. To make a long story short, after much fiddling with recursion and induction we can establish the field structure on $\mathbb{R}_C$ and prove that it is indeed the initial Cauchy-complete field.
2. To see the proof, consult HoTT book Theorem 11.3.50.

What about computable mathematics?

- Computable models:
  - Truncation make sure there is no higher homotopy.
  - Realizability models, and in particular TTE, should model such truncated types.
  - So this would give computability for one part of HoTT.
- Other examples:
  - Sierpinski space Σ
  - Borel sets
  - ...

1. In conclusion, let me say a bit about why I think this is useful for computable analysis.
2. HoTT itself *is* a model of computability by work on cubical sets by Bezem et. al. However, the cubical sets are fairly involved and it may be worthwhile looking for computable models of fragments of HoTT, and in particular the truncated fragment we used. And here realizability models should work, TTE is one of them. I am convinced that if someone's student worked out the details, they would discover that $\mathbb{R}_C$ is the usual admissible representation of reals.
3. There are perhaps some ideas about implementation in the construction, but I am not reallly sure.
4. Lastly, the real numbers are just one possible construction. At the recent TYPES meeting a construction of the Sierpinski space Σ was given by Altenkirch and Danielsson as a higher inductive-inductive type. Once you have Σ, you can start doing topology.
5. There are other examples that nobody has looked at. For instance, the Borel sets ought to be describable in terms of infinitary suprema and infima with governeing equations. This should allows us to do some measure theory, perhaps.

This material is based upon work supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF under Award No. FA9550-14-1-0096. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Air Force Office of Scientific Research, Air Force Materiel Command, USAF.